Software Defect Repair Times: A Multiplicative Model

Swapna S. Gokhale
Dept. of CSE
Univ. of Connecticut
Storrs, CT 06269
01-860-486-2772
ssg@engr.uconn.edu

Robert E. Mullen
Corporate Quality Metrics
Cisco Systems, Inc.
Boxborough, MA 01719
01-978-936-1585
bomullen@cisco.com

ABSTRACT

We analyzed over 10,000 software defect repair times collected for nine products at Cisco Systems, to confirm our hypothesis that software defect repair times can be characterized by the Laplace Transform of the Lognormal (LTLN) distribution. This hypothesis originated from the observation that software defect repair times are influenced by the multiplicative interplay of several factors. The Lognormal distribution is a natural choice to model rates of occurrence of such phenomenon. Conversion of the Lognormal rate distribution to an occurrence time distribution yields the LTLN. Our results also confirm that the LTLN distribution provides a statistically better fit to the observed repair times than either of the two most widely used repair time distributions, the lognormal and the exponential.

Categories and Subject Descriptors

D.2.8 SOFTWARE ENGINEERING, Metrics, Process metrics

General Terms

Reliability.

Keywords

Lognormal, Multiplicative factors, Software repair

1. INTRODUCTION

The consideration of defect repair rates and repair times is as important as defect failure rates and failure times when devising ways to improve software reliability and availability. For example, when a defect manifests as a failure in the field, it is necessary to repair it as soon as possible to minimize the down time, the number of service interruptions, or the duration of reduced functionality. Further, even for defects that are discovered internally prior to release, during the development and testing phases, shorter repair times reduce time-to-market and improve the quality at initial release. These considerations motivate proposing a defect repair model, determining its functional form, and empirically validating that hypothesis.

Despite the strong influence of repair times on software reliability, schedules and resources, very few research efforts study the causes of software defect repair times. Of these, most of the efforts incorporate repair times into software reliability models [11,27,28], without explicitly characterizing the defect repair times. A handful of efforts which provide a mathematical treatment of defect repair times in software assume that they follow either an exponential or a lognormal distribution. While some of these efforts offer empirical evidence, most do not provide any theoretical justification for this assumption [25,30].

In contrast, we observe that software defect repair times are influenced by several factors which capture pertinent traits of the defect itself, the personnel assigned, and the development and maintenance process. We discuss a set of representative factors and hypothesize that their interplay may lead to a lognormal distribution of defect repair rates. Our hypothesis is grounded in preliminary observations made directly from the data [25,21], theoretical evidence and a number of significant recent results which suggest that the rates of other events (failure rates of faults, execution rates of code) in software approach the lognormal [3,9,19,20].

Although a formal model of defect repair rate can provide some insights, it is difficult to observe repair rates in practice. A metric which can be observed in practice, along with its impact on reliability and development schedule, is defect repair time. It is thus necessary to convert from a *repair rate* to a *repair time* distribution. In this paper, we interpret repair time as the duration of the time from when the defect is written until the defect is repaired. It does not include either the time to write the defect or the time to distribute the repair or place it in service. It also does not include the provisioning of work-arounds and avoidances which may rapidly resolve a customers' problem. We convert from a repair rate to a repair time distribution using the Laplace transform, since the mathematical transformation from a rate to an occurrence time distribution is equivalent to the Laplace Transform of the rate distribution [19]. We refer to this as Laplace Transform of lognormal (LTLN) repair time distribution. We analyze empirical repair times collected for several products at Cisco for evidence of the LTLN, exponential and lognormal distributions, the latter two being the most commonly used repair time distributions [25], and compare the quality of fits using log-likelihood and AIC [26]. Our results indicate that the LTLN distribution is significantly more likely to generate the observed repair times than the exponential and the lognormal distributions.

The layout of this paper is as follows: Section 2 reviews the conditions that generate the lognormal distribution, and discusses how those conditions may be met by typical software defect repair processes. Section 3 defines the LTLN repair time distribution. Section 4 describes the data collection and analysis. Related

PROMISE'08, May 12–13, 2008, Leipzig, Germany.

Table 1: Factors affecting software defect repair rates

Defect						Personnel				Process Support			
Severity		Clarity		Difficulty		Speed		Skills		Resources		Tools	
Prob.	Value	Prob.	Value	Prob.	Value	Prob.	Value	Prob.	Value	Prob.	Value	Prob.	Value
0.02	1	0.20	1	0.10	0.5	0.05	0.5	0.20	0.6	0.40	1	0.15	0.8
0.19	1.78	0.30	1.5	0.40	1	0.45	1	0.30	0.9	0.30	1.5	0.35	0.9
0.79	3.49	0.30	2.5	0.30	2	0.30	3	0.30	1.2	0.20	2	0.35	1.1
		0.20	4	0.20	3	0.20	10	0.20	1.7	0.10	4	0.15	1.4
Var.	0.094	Var.	0.232	Var.	0.305	Var.	0.876	Var.	0.121	Var.	0.178	Var.	0.077

Var. is the variance of the natural log of each factor.

research is in Section 5. Section 6 offers concluding remarks and directions for future research.

2. ORIGIN OF REPAIR RATES

In this section we provide an overview of the lognormal distribution. We then discuss the rationale governing the genesis of lognormal repair rates.

2.1 Overview of the lognormal distribution

The lognormal distribution is well known in many fields, including ecology, economics and risk analysis [7]. It arises in a natural way when the variants value is determined by the multiplicative effect of various random factors, just as the normal distribution arises as a consequence of the additive effect of many random factors [13]. The multiplicative form of the Central Limit Theorem (CLT) tells us that under very general conditions the distribution of the products of those factors is asymptotically lognormal, similar to the way in which the additive form generates a normal distribution.

Two forms of the CLT presented in Aitchison and Brown's monograph [2], state conditions under which a product of random variables is asymptotically lognormal. The Lindeberg-Levy form of the CLT requires that all the factors are from the same distribution, but the Lyapunov form removes that condition and adds one on the expectation of their third moment. There are other more general extensions of the CLT which remove or adjust these restrictions. For example Petrov [24] provides additional theorems in which the variables are not necessarily identically distributed. Loeve [17] establishes alternatives to the assumption of independence. Patel and Read [23] outline further extensions. In summary, key assumptions which lead to the lognormal are the multiplicative effect of many relatively independent random factors, with no one factor effectively dominating the others. We now discuss some of typical factors that may influence the defect repair rate and how their multiplicative impact gives rise to a lognormal.

2.2 Multiplicative factors in repair

We propose that several factors, describing the characteristics of the defect, the personnel, and the supporting process influence the rate at which a software defect is repaired. A few such representative factors are summarized in Table 1. For each factor, we approximate a continuous distribution with a discrete one taking three or four levels. The probability and the rate adjustment for each level are presented in the two sub-columns for each factor. The rate adjustment indicates how the relative defect repair rate should be scaled for that level. These probabilities and rate adjustments are drawn from prior research and from anecdotal evidence.

These factors, along with their specific values, ranges or weightings are chosen solely for the sake of illustration and discussion. We note that they are by no means universally applicable, and other factorizations are certainly possible. For example, the factors from the effort prediction data sets available at www.promisedata.org would be other candidate factorizations. We hypothesize that the collective impact of factors like these on the repair rate of a defect is multiplicative, consistent with the COCOMO model in which the multiplicative impact of the individual cost drivers determines the ultimate scaling factor used to scale the estimated nominal effort [4,5]. Our main objective in the discussion of these factors is to illustrate how the various factors interacting multiplicatively can lead to a lognormal distribution of repair rates. A secondary objective is to show how a set of plausible factors leads to the observed spread in the repair rates.

A discussion of these factors follows:

Severity: This factor considers the influence of defect severities on repair times. Consistent with the classification in many software organizations, defects are categorized into three severity levels. The probabilities and rate adjustments for the three levels are from our earlier work [10] which studied resolution times, finding that the more severe defects are resolved more rapidly.

Clarity: A customer typically interacts with a support engineer to report a defect encountered in the field, however, internally found defects are usually reported by the finder. The defect's symptoms and the conditions under which it occurs are documented in the defect tracking system A well-documented defect has a higher chance of being reproduced and repaired quickly than a poorly documented one. In fact, an extremely well-documented defect may be accompanied by a suggested solution. The factor clarity, with its four levels, is intended to capture this impact on repair times. These four levels, along with their probabilities and rate adjustments are based on an informal survey.

Difficulty: This factor accounts for the inherent hardness of the repair including the depth and the breadth of the technical problem needing understanding, its magnitude, and the presence of constraints on the solution. It includes the eight COCOMO cost drivers related to Platform and Product, namely Complexity, Volatility and constraints with respect to Reliability, Storage, Database size, Execution time, Reusability, and Documentation [5].

Speed: This factor captures the contribution of innate aptitude to the performance of a programmer. Prior research and other evidence

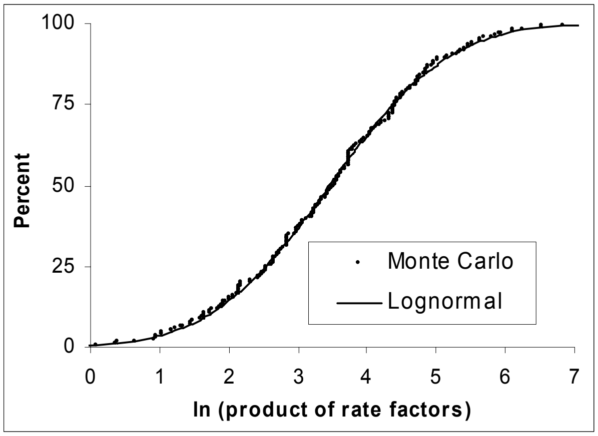

Figure 1: Monte Carlo generated using seven factors compared with fitted Lognormal CDF.

suggest that there is significant variability in individual programmer productivity and speed [14].

Skills: This factor reflects the influence of the programmer performance arising from training and experience in the specific technology, component and the feature of the defect. It encompasses the four COCOMO cost drivers of Personnel continuity, Applications experience, Platform experience, and Language and Tools experience [5].

Resources: This factor accounts for the availability of sufficient specific hardware platforms and other equipment, first necessary to reproduce the problem and then to verify its repair. Since in most software organizations which develop large-scale products there is a fair chance of resources being available, we assume the likelihood of this best-case scenario to be 40%. We expect that more severe resource shortages are increasingly improbable.

Tools: The use of tools to support and automate the activities associated with repair will impact the repair rate. These activities may include bookkeeping and documentation including defect tracking, understanding and reproducing the defect, and ensuring the validity of the fix through regression testing. The levels and rate adjustments for this factor are adapted directly from the COCOMO model, where in the best case, superior tool support speeds up repair by a factor of 1.2, whereas in the worse case where many of the activities need to be performed manually, slows down repair by a factor of 1.4 [4].

In Figure 1 we compare the rate distribution obtained from Monte Carlo simulation (drawing 250 sets of 7 random factors based on Table 1) with a fitted lognormal distribution. The standard deviation obtained from simulation is 1.36, reflecting the combined effect of the factors. As expected this is very close to 1.372, the standard deviation computed from the square root of the sum of the variances of the natural logarithms of the factors (summarized in the last row in Table 1). More significantly, the excellence of the visual fit indicates the proposed repair model, using a small number of commonly used factors has the potential to generate a lognormal distribution of rates.

3. REPAIR TIME HYPOTHESIS

In this section we present the mathematical formulation of the LTLN repair time hypothesis, which is derived from the lognormal rate distribution motivated in Section 2. We also provide an interpretation of the model parameters.

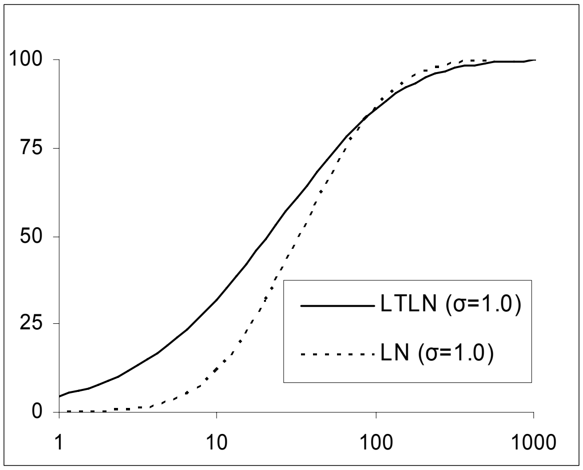

Figure 2: x-axis is days. CDF showing difference between the LN, and LTLN for σ=1.0, μ = -3.5.

3.1 Mathematical formulation

We assume that each defect in a given product family has a characteristic repair rate, λ, drawn from a lognormal distribution which has parameters determined by factors like those considered in Section 3. To say that the distribution of the repair rates of software defects is lognormal is to say that the logarithms of the repair rates ln(λ), follow the Gaussian or normal probability distribution function (pdf). For $(\lambda > 0)$:

$$dL(\lambda) = \frac{1}{\lambda\sigma\sqrt{2\pi}} e^{-(\ln(\lambda)-\mu)^2/2\sigma^2} d\lambda \qquad (1)$$

For the lognormal, the mean, median, and mode of the log-rate are identical and equal to μ. The variance of the log-rate is σ^2. The mean rate is exp ($\mu+\sigma^2/2$), the median rate is exp (μ), and the mode is exp ($\mu-\sigma^2$).

To derive the LTLN repair time model, we further hypothesize the time to repair of a defect, that is the age at fix, is an exponentially distributed random variable determined by its rate, the rate itself being a random variable depending on the parameters of the underlying lognormal. Thus, this is a doubly stochastic model as described by Miller [18]. Informally, this represents the fact that knowing all the factors which may influence the repair of a defect (severity, skills, etc) does not allow an exact determination of the repair time because there are additional extrinsic considerations that may lead to faster or slower outcomes in each case.

Since the repair time is exponentially distributed conditional to a given repair rate λ, the probability that the defect of rate λ is not repaired until time t or later is exp ($-\lambda$t). The probability that the defect was repaired before time *t* is then 1 - exp($-\lambda$t). Since λ is distributed as L($\lambda|\mu$, σ^2) then *M(t)*, the cumulative distribution function of the time taken to repair a defect is given by:

Table 2: AGE series for product families

AGE (days)	B	C	G	M	N	S	T	U	Y
0	118	123	138	227	100	171	122	143	175
1	230	207	218	403	176	301	258	281	343
2	288	257	263	505	223	375	345	361	445
3	343	300	304	583	293	427	433	418	523
4	384	345	345	639	349	481	498	467	583
5	409	394	368	714	402	534	552	529	657
6	459	449	401	777	446	574	600	595	728
7	507	509	435	853	498	628	657	631	810
8	550	539	464	898	532	668	716	680	864
9	577	565	488	935	561	694	736	699	915
11	598	605	522	1005	611	741	782	739	995
13	650	641	550	1063	661	784	839	776	1077
15	697	691	575	1133	723	824	889	820	1143
18	738	727	601	1200	772	858	921	861	1204
21	778	777	625	1256	824	903	977	894	1274
25	816	814	648	1326	873	948	1024	938	1319
29	852	846	688	1384	919	985	1067	970	1374
34	881	877	718	1433	968	1023	1091	996	1422
40	929	909	748	1469	1005	1055	1126	1024	1477
47	961	933	785	1510	1058	1086	1150	1057	1522
55	990	952	814	1538	1090	1105	1186	1079	1555
64	1017	967	839	1572	1113	1125	1204	1100	1610
74	1038	985	864	1610	1138	1132	1229	1111	1643
86	1054	1001	883	1642	1168	1140	1255	1136	1666
99	1064	1017	903	1676	1185	1147	1276	1157	1686
114	1075	1028	926	1712	1204	1158	1298	1179	1695
132	1082	1036	956	1731	1228	1162	1314	1199	1714
152	1086	1048	969	1748	1250	1176	1332	1209	1730
175	1093	1058	1000	1762	1269	1181	1356	1213	1734
202	1097	1063	1020	1784	1277	1189	1380	1214	1739
233	1102	1071	1039	1795	1291	1200	1408	1216	1744
268	1103	1074	1058	1804	1295	1205	1435	1218	1749
309	1105	1080	1073	1817	1303	1215	1449	1220	1753
356	1107	1081	1086	1828	1313	1230	1460	1224	1763
410	1112	1085	1099	1833	1319	1236	1480	1225	1765
Total	1125	1096	1139	1858	1340	1270	1497	1226	1775
Mean AGE	30.733	32.400	69.709	37.985	44.363	43.865	48.787	24.011	27.431
s.d. AGE	73.560	75.925	125.413	81.797	88.519	114.394	91.970	50.578	61.679

$$M(t) = 1 - \int_{\lambda=0}^{\infty} \exp(-\lambda t)\, dL(\lambda) \qquad (2)$$

This integral is formally equivalent to the Laplace transform of the lognormal and has no simple form. Clearly $M(0)=0$, and $M(\infty)=1$, which is to say that the probability of a defect being repaired before time zero is zero, but ultimately, the defect will be repaired. (The integral is also encountered when transforming a distribution of failure rates to a failure time distribution [19]). The intractable integral is computed numerically by changing variables so the integral is of the standard normal distribution and computing its height at regular intervals. The mean repair time can be computed once the integral is numerically computed. A detailed discussion of this approach can be found in [19].

Figure 2 illustrates the difference between a lognormal distribution and the LTLN. Although both are based on lognormal of the same σ , the LTLN is broader. Near zero the lognormal is zero, but the LTLN pdf (not shown) is not.

3.2 Interpretation of lognormal parameters

Conceptual advantages of the lognormal include the relative simplicity of its parameters and the way it links the observed properties of various factors influencing repair to the repair rates.

The parameter σ makes the greatest qualitative difference and allows the lognormal its flexibility. σ, the standard deviation of the log rates, increases as the disparity in the rate adjustments between the best and worst levels of each factor increases. If σ is zero, all defects have the same repair rate, regardless of their difficulty, severity, and clarity and the experience and the skills of the engineers fixing them. It is doubtful that in practice, all defects are equivalent with respect to these parameters. But if that were the case, then we would observe repair times following an exponential distribution. If σ is 2.0, the ratio of rates between the second percentile and the ninety-eighth percentile exceeds exp(8), a factor of approximately 3000.

μ has a simple interpretation: if rates are plotted on a log scale, changing μ merely moves the distribution to the right or left. μ may be changed by changing the repair rates of all defects by a constant factor; for example, a process speedup or using different units of time. For μ = -2, the median rate is exp(-2) or .14 per day and approximately half will be repaired after five days for all σ.

Changing either μ or σ — both of which relate to ln(rate) — does not affect the other. However, changing either μ or σ affects *both* the mean and variance of the rates themselves.

4. DATA EXTRACTION AND ANALYSIS

In this section we first describe the extraction and processing of the raw data. We then describe our analysis of the data using the LTLN, LN and EXP models and compare the results.

4.1 Data description

The defect tracking system records the date on which each defect was created. The system also tracks the progress and the status of a defect, which is given by the current “state” of the defect. A detailed description of the various different states in the life of a defect, the corresponding state transitions and the considerations and circumstances which cause these state transitions are described in our earlier work [10].

Broadly, the states of a defect can be classified into three categories. The first category consists of transitory states where the defect is in the process of being in-queue, analyzed, and resolved. The second category covers includes terminal states which are reached if the defect is either a duplicate of another defect, or is spurious, or it cannot be reproduced, or is designated as resolved without repair. The third category includes terminal states in which the defect is resolved through repair and a bug-fix is applied to the code. For the purpose of this paper, we consider only defects that belong to the third category, i.e., those that were resolved via repair.

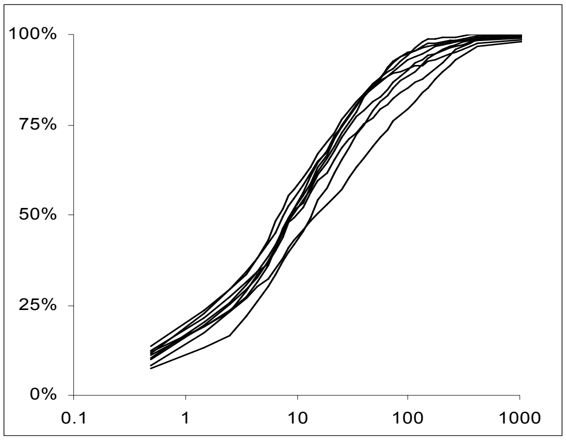

Figure 3: Repair AGE CDFs for nine products (x-axis in days)

Defect repair data were extracted for nine product families, over periods ranging from one month to one year. Defects selected were those for which repair was completed during the considered period. The defects may have been created at any earlier time. For each product family, the duration of the considered period was chosen such that at least 1000 defect repairs were available.

The AGE function of the defect tracking application provided the number of days between the defect creation and repair. Defects fixed on the same day as they arrived have an AGE of zero, ones fixed on the following day have an AGE of one, and so on. These times are not representative of customer experience which also includes the receipt of of workarounds, avoidances, fixes already underway.

The sequences of AGES along with the cumulative defect counts for the nine product families are provided in Table 2. In Table 2, for example, for product B, 118 defects have an AGE of zero, 230 defects haves AGES less than or equal to 1 and so on. Also listed in Table 2 are the total number of defects considered for each product family (third row from the bottom), and the mean and the standard deviation of the AGE of the defects (the second row from the bottom and the last row). For each product family it can be seen that the standard deviation of the AGE is significantly greater than the mean AGE. Figure 3 shows the empirical distributions are highly skewed, having a long tail to the right.

4.2 Data processing and aggregation

A defect repaired on the same day as it arrived appears as AGE zero in our data, though in fact it was open for a fraction of a day. Defects with AGE of zero, in fact of any AGE need their AGES adjusted to eliminate this anomaly. For fitting models we consider AGE zero defects to represent defects actually fixed within the range 0 to .5 days, AGE one defects to include defects fixed within .5 to 1.5 days, and so on.

We aggregated the data by partitioning the sequence into 36 ranges. AGES less than ten days were not combined, preserving resolution

Table 3: Model comparison for product families

Product Family	LTLN Neg. LLH	LN Neg. LLH	EXP Neg. LLH	AIC LTLN vs LN
B	142.71	159.04	517.66	32.67
C	138.64	155.82	526.54	34.35
G	141.57	149.19	703.69	15.25
M	148.48	165.46	982.63	33.97
N	133.54	155.06	529.42	43.03
S	142.53	146.89	983.76	8.72
T	165.01	169.64	869.63	9.26
U	138.93	154.94	489.80	32.02
Y	152.23	182.55	649.73	60.65

for the lowest, most heavily populated AGES. The latter ranges are of proportionately increasing length so there are a sufficient number of defects in each range for statistical analysis.

4.3 Comparative analysis

Repair times are commonly modeled by the exponential (EXP) and lognormal (LN) distributions [30]. A recent study [25] confirmed that LN is superior to both Weibull and gamma distributions for software and hardware repair times. Because of their previous success and simplicity, respectively, we consider the primary competing models to be LN and EXP.

For each product family, the maximum likelihood estimates of the parameters of the LTLN, LN, and EXP models were obtained. For objective comparison, we computed the log-likelihood of the observed data being generated by each model. However, the models cannot be compared directly using log-likelihoods because they have different numbers of parameters. Therefore, we compare the models using the Akaike Information Criterion (AIC), which is similar to a likelihood ratio test, but penalizes the models, here the LTLN and the LN which more parameters than the EXP model. The AIC for each model is computed as [26]:

AIC = - 2 * log_likelihood + 2 * num_parameters

The model with the lower value is the better. Two units of AIC is significant, four very significant [1].

Table 3 provides the (negative) log-likelihoods of the three models for the nine product families. We see directly the exponential model has far less likelihood of generating the observed data than either the lognormal or LTLN and we will not consider it further. For the LN and the LTLN we provide the difference between AIC values. In every case the evidence is very significantly in favor of the LTLN hypothesis, indicating that the LTLN is the best model of the underlying repair process for each family. Since these product families are independent, we consider this as strong evidence in favor of the LTLN hypothesis.

Table 4: Model parameters for product families

Product Family	LTLN		LN	
	σ	μ	σ	μ
B	1.350	-2.602	1.883	2.024
C	1.375	-2.625	1.897	2.050
G	2.028	-3.127	2.443	2.574
M	1.631	-2.616	2.082	2.050
N	1.301	-3.050	1.844	2.476
S	1.781	-2.462	2.167	1.909
T	1.619	-2.867	2.036	2.312
U	1.365	-2.389	1.867	1.818
Y	1.256	-2.586	1.804	2.009

The parameters of both the LN and the LTLN models which provide the best fit to the observed AGES are summarized in Table 4. The values of σ are similar with that of the LN being consistently larger. If σ were zero it would be equivalent to exponential repair times. On the other hand, the values of μ are of opposite sign, with those for LTLN being more negative than LN. Both patterns are results of fitting a lognormal to an underlying distribution of *rates* in the LTLN case and fitting a lognormal directly to the *times* in the LN model. Because the rates and times are approximately inversely related, the values of μ in the LTLN and LN models are similar but of opposite sign. For the same reason, the values of σ are related. For the LTLN the fact that σ ranges between 1.0 and 2.0 suggests that the process is roughly similar across the nine product families.

5. RELATED RESEARCH

A number of research efforts have focused on incorporating explicit repair into software reliability growth models, with the intent to relax the simplifying assumption of instantaneous repair, and hence providing realistic estimates of reliability. Most of these efforts assume the repair rate to be constant (repair time to be exponential) [8,15,16,28,27]. Gokhale et al. [11,12] incorporate various repair policies into software reliability models.

In most of these efforts, the repair time distributions are assumed without any theoretical justification or empirical evidence. The work closest to ours is in [25] in which the LN distribution is used to model software defect repair times. However, empirical data analysis for the products considered in this paper categorically rules out the LN distribution in favor of the LTLN distribution to characterize software defect repair times.

An approach to predicting software defect repair times is presented by Zeller et al. [29]. It relies on mining issue reports in bug databases for similar issues and using their average repair times for prediction. Their approach is purely data-oriented and uses past observed defect repair times to predict the future ones based on the similarity of issues. It does not attempt to relate the observed repair times to the characteristics of the defect resolution process. By contrast, our approach provides a mathematical foundation to characterize repair times and to understand and justify them in terms of the pertinent aspects of the resolution process. A significant advantage of a mathematical foundation is that it can offer insights into how the process can be improved and can readily enable predictive or "what-if" analysis of the impact of different improvement strategies.

6. CONCLUSIONS

Prior research has often assumed that software defect repair times follow an exponential or lognormal distribution, but generally not explained why or provided any detailed evidence. In this paper, we proposed a multiplicative model which generates a lognormal distribution of defect repair rates and converted the rate distribution to LTLN repair time distribution. We then analyzed over 10,000 defect repair times for each of nine products using the LTLN, exponential and lognormal distributions. A comparison of the results indicated that the LTLN distribution was significantly more likely to generate the observed repair times than the lognormal and exponential distributions. The paper thus utilizes significant industrial data to confirm a research hypothesis linking software repair processes to a mathematical model.

This opens several areas for further progress. It would be advantageous to further our understanding of the factors which lead to lognormal repair rates and then to exploit this understanding to enhance software reliability by evaluating and implementing maintenance process improvements. Second, the relationship between the exponential, LTLN, and lognormal distribution needs to be investigated. Third, to the extent failure rates and repair rates both follow the lognormal distribution, it may be possible to create joint process models of failure and repair with greater conceptual unity. Finally, the LTLN repair time hypothesis needs to be evaluated in other applications besides software reliability.

7. ACKNOWLEDGMENTS

We thank Cisco management, especially David Hsiao, for their encouragement. The research at Univ. of Connecticut was supported in part by a CAREER award (#CNS-0643971) from the National Science Foundation.

8. REFERENCES

[1] Akaike H. 1982. Prediction and Entropy, MRC Technical Summary Report #2397. NTIS.

[2] Aitchison J. and Brown J.A.C. 1969. The Lognormal Distribution, Cambridge University Press, NY.

[3] Bishop P. and Bloomfield R. 2003. Using a Log-normal failure rate distribution for worst case bound reliability prediction. In Proceedings of the Fourteenth International Symposium on Software Reliability Engineering (Denver, CO, November 2003). 237-245.

[4] Boehm B. 1981. Software Engineering Economics, Prentice Hall.

[5] Boehm B. 2000. Software Cost Estimation with COCOMO II, Prentice Hall.

[6] Chillarege R. 1996. Orthogonal Defect Classification, in M.R. Lyu (ed.). Handbook of Software Reliability Engineering, McGraw-Hill.

[7] Crow E.L. Crow and Shimizu K. ed. 1988. Lognormal Distributions: Theory and Applications. Marcel Dekker, NY.

[8] Dalal S. R. and Mallows C. L. 1990. Some graphical aids for deciding when to stop testing software. IEEE Transactions on Software Engineering. 8, 2 (February 1990), 169-175.

[9] Gokhale S. and Mullen R. 2004. From test count to code coverage using the Lognormal. In Proceedings of the Fifteenth International Symposium on Software Reliability Engineering (St. Malo, France, November 2004). 295-304.

[10] Gokhale S. and Mullen R. 2006. Queuing models for field defect resolution process. In Proceedings of the Seventeenth International Symposium on Software Reliability Engineering (Raleigh, NC, November 2006). 353-362.

[11] Gokhale S., Lyu M. R. and Trivedi K. S. 2004. Analysis of software fault removal policies using a non homogeneous continuous time Markov chain. Software Quality Journal (September 2004), 211-230.

[12] Gokhale S., Lyu, M. R., and Trivedi K. S. 2006. Incorporating fault debugging activities into software reliability models. IEEE Transactions on Reliability. 51, 2 (December 2006), 578-590.

[13] Johnson N. L., Kotz S., and Balakrishnan N. 1994. Continuous Univariate Distributions, vol. 1, Wiley, New York.

[14] Keyes J. 2003. Software Engineering Handbook, CRC Press.

[15] Kremer W. 1983. Birth and death bug counting. IEEE Transactions on Reliability. R-32, 1 (April 1983), 37-47.

[16] Levendel Y. 1990. Reliability analysis of large software systems: Defect data modeling. IEEE Transactions on Software Engineering. 16, 2 (February 1990), 141-152.

[17] Loeve M. 1963. Probability Theory. Van Nostrand.

[18] Miller D. R. 1985. Exponential order statistic models of software reliability growth. NASA Contractor Report 3909. NTIS.

[19] Mullen R. E. 1998.The Lognormal distribution of software failure rates: Application to software reliability growth modeling. In Proceedings of the Ninth International Symposium on Software Reliability Engineering (Paderborn, Germany, November 1998). 134-142.

[20] Mullen R. E. 1998. The Lognormal distribution of software failure rates: Origin and evidence. In Proceedings of the Ninth International Symposium on Software Reliability Engineering (Paderborn, Germany, November 1998). 124-133.

[21] Mullen R. E. 2006. Characterizing Software Defect Repair Times. Proceedings of the Seventeenth International Symposium on Software Reliability Engineering (Raleigh, NC, November 2006). Industry Practices.

[22] Mullen R. E. and Gokhale S. 2005. Software defect rediscoveries: A discrete Lognormal model. In Proceedings of the Fifteenth International Symposium on Software Reliability Engineering (Chicago, IL, November 2005). 203-212.

[23] Patel J. K. and Read B. C. 1996. Handbook of the Normal Distribution. Marcel Dekker.

[24] Petrov V. V. 1975. Sums of Independent Random Variables. Springer-Verlag, (tr. A.A. Brown).

[25] Schroeder B. and Gibson G. A. 2006. A large-scale study of failures in high-performance computing systems. In

Proceedings of the International Conference on Dependable Systems and Networks (Philadelphia, PA, June 2006).

[26] Sakamoto Y., Ishiguro M., and G. Kitagawa. 1986. Akaike Information Criterion Statistics. D.Reidel, Holland.

[27] Schneidewind N. F. 2002. An integrated failure detection and fault correction model. In Proceedings of the International Conference on Software Maintenance (December 2002). 238-241.

[28] Stutzke M. A. and Smidts C. S. 2001. A stochastic model of fault introduction and removal during software development. IEEE Transactions on Reliability. 50, 2 (June 2001), 184-193.

[29] Weiss, C., Premraj, R., Zimmermann, T. and Zeller, A. 2007. How long will it take to fix this bug? In Proceedings of the Fourth International Workshop on Mining Software Repositories (2007).

[30] NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook /apr163, July 18, 2006.